

Foundations of the Socio-Physical Model of Activities (SOMA) for Autonomous Robotic Agents¹

Daniel Beßler^a, Robert Porzel^b, Mihai Pomarlan^c, Abhijit Vyas^a, Sebastian Höffner^b, Michael Beetz^a, Rainer Malaka^b, and John Bateman^c

^aDepartment of Artificial Intelligence, University of Bremen, 28359 Bremen, Germany

^bDigital Media Lab, TZI, University of Bremen, 28359 Bremen, Germany

^cDepartment of Linguistics, University of Bremen, 28359 Bremen, Germany

Abstract. In this paper, we present foundations of the Socio-physical Model of Activities (SOMA). SOMA represents both the physical as well as the social context of everyday activities. Such tasks seem to be trivial for humans, however, they pose severe problems for artificial agents. For starters, a natural language command requesting something will leave many pieces of information necessary for performing the task unspecified. Humans can solve such problems fast as we reduce the search space by recourse to prior knowledge such as a connected collection of plans that describe how certain goals can be achieved at various levels of abstraction. Rather than enumerating fine-grained physical contexts SOMA sets out to include socially constructed knowledge about the functions of actions to achieve a variety of goals or the roles objects can play in a given situation. As the human cognition system is capable of generalizing experiences into abstract knowledge pieces applicable to novel situations, we argue that both physical and social context need be modeled to tackle these challenges in a general manner. The central contribution of this work, therefore, lies in a comprehensive model connecting physical and social entities, that enables flexibility of executions by the robotic agents via symbolic reasoning with the model. This is, by and large, facilitated by the link between the physical and social context in SOMA where relationships are established between occurrences and generalizations of them, which has been demonstrated in several use cases in the domain of everyday activities that validate SOMA.

Keywords. design patterns, domain ontology, autonomous robotics

1. Introduction

In spite of undoubtedly being ubiquitous, the domain of *everyday activities* poses considerable challenges. Many people perform activities such as cooking or cleaning almost every day. This includes to select and manipulate ingredients, use tools and

¹This work was funded by the German Research Foundation (DFG) as part of Collaborative Research Center (CRC) 1320 EASE – Everyday Activity Science and Engineering, University of Bremen (<http://www.ease-crc.org/>), subprojects H2, P1 and R1.

devices, arrange the prepared dishes for serving and clean up afterwards – and do it all quick and robust without recourse to an advanced computational theory. Further, the amount of information provided in a description of a task – such as a natural language command requesting its completion – is much less than the amount of information needed to perform the task. This raises the question how humans are able to decide so quickly what to do next, despite ambiguity and underspecification.

Lenat and Feigenbaum observe that “more knowledge implies less search” [1]. Knowledge of many possible plans, as well as knowledge of the world in general, seems to be the secret of human performance. There is no algorithmic reason why tomatoes and oregano go well together, or why a raw egg must be handled with care. The cook simply has to know these things. Such knowledge of the world is taught, observed, and then ingrained by practice. As Anderson observes, “an agent has a great deal of knowledge [of everyday activities], which comes as a result of the activity being common” [2]. As human beings, we acquire such knowledge naturally over the course of our lives. A lot of what we learn, we learn by doing, or by watching others. This suggests that a robot must have mechanisms to organize and interpret observations, either of its own behavior or of other agents, into structures that are then amenable for other computational tasks. To this end, the concept of *narratively-enabled episodic memories* (NEEMs) was introduced [3,4]. NEEMs are comprehensive logs of raw sensor data, actuator control histories and perception events, all semantically annotated with information about what the robot is doing and why using the terminology provided by SOMA.

The computational tasks that must be solved when acting in the physical world are often very complex and beyond what is thought to be tractable. This, however, is only the case when these problems are regarded in their full generality and not for restricted versions of these problems. However, the knowledge representing such pragmatic solutions goes beyond modeling physical events and requires models of the social context by means of which the physical events can be realized and interpreted. For this, we employ an existing upper-level ontology that we augment with general design patterns and specific modules that are pertinent for robot knowledge modeling. In this paper we provide an overview, of this approach for robot knowledge modeling where all of our extensions to the given foundational framework rely on the differentiation between the observable physical domain and the conceptualized social interpretations thereof.

The overall goal of our research is to enable robotic agents to perform everyday activities with similar robustness and flexibility as human agents do. Given this aim, we must, in some sense, get the robot to know what humans know about the world, at least as it pertains to everyday activities. This presents several challenges, beyond the scope of what needs to be known to represent such intricate and extensive domain. There lies the question of how to represent and structure this knowledge in order to realize a similar robustness, flexibility and efficiency in performance. In addition, there are challenges concerning the acquisition and learnability of the corresponding structures. In this paper, we will focus on how this knowledge is represented. We will describe our employment of an existing upper level ontology and the development of several ontology modules aimed to address this general ontology design challenge. The resulting ontology is openly available ², and additional documentation is available online ³. As depicted in Figure 1,

²<https://github.com/ease-crc/soma>

³<https://ease-crc.github.io/soma/>

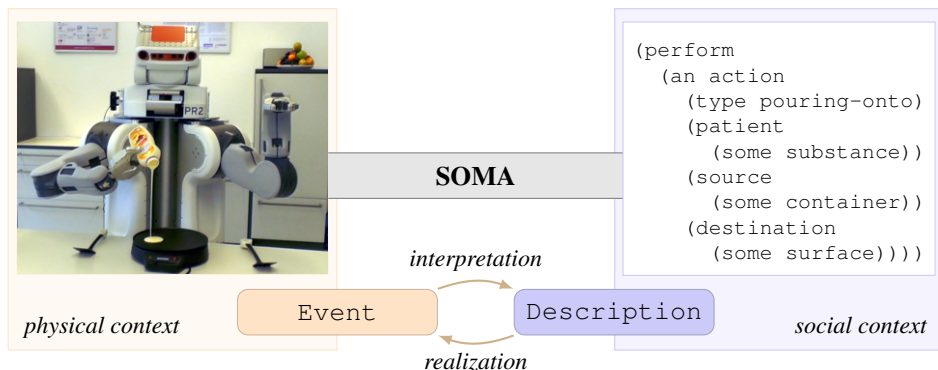


Figure 1. SOMA represents physical and social context, and supports robotic agents in interpreting observed events, and realizing abstract descriptions.

our focus lies on representing both the physical context of realized everyday activities, as well as interpretations thereof as the social context.

2. Related Work

Ontology-based knowledge representation and reasoning in autonomous robot control is a fairly extensive field of research with developments in both service and industrial robotics. In the following, we will briefly discuss the most relevant works. A more detailed discussion about how ontologies are used to support robot autonomy is provided by Olivares et al. [5].

One example in the industrial robotics domain is the ROSETTA project [6,7]. Its initial scope was reconfiguration and adaptation of robot-based manufacturing cells, however, the authors have, since then, further developed their activity modeling for coping with a wider range of industrial tasks. Other authors have focused on modeling industrial task structure, part geometry features, or task teaching from examples [8,9,10,11]. Compared to the everyday activity domain, industrial tasks considered in above works are more structured, and less demanding in terms of flexibility.

An approach to activity modeling in the service robotics domain is presented by Tenorth and Beetz [12]. Foundationally, their modeling is based on a subset of the discontinued OpenCyc ontology [13] with much weaker axiomatization compared to our foundational layer, and less inferential power and guidance during modeling. The scope of their work is similar to ours as the authors also consider how activity knowledge can be used to fill knowledge gaps in abstract instructions given to a robotic agent performing everyday activities. However, the scope of the work presented here is wider, as we also consider how activity knowledge can be used for the interpretation of observations. Our activity modeling is further more detailed in terms of activity structure as we also consider the processes and states that occur during an activity. Another difference is that, in their modeling, there is no distinction between physical and social context, but this dichotomy is central in SOMA.

A more general approach to activity modeling for robotic agents is presented by the IEEE-RAS working group ORA [14]. The group has the goal of defining a

standard ontology for various sub-domains of robotics, including a model for object manipulation tasks. It has defined a core ORA ontology [15], as well as additional modules for industrial tasks such as kitting [16]. In terms of methodology, we differ in foundational assumptions we assert, which has important consequences on the structure of our ontology, modeling workflow, and inferential power. In the case of ORA, the SUMO upper-level ontology is used as foundational layer. However, the foundational layer of SUMO is rather weakly axiomatized compared to other models. In particular central in SOMA is the distinction between ground and descriptive concepts to represent physical and social activity context, and that this distinction is tightly coupled with the foundational layer.

3. Overview

In this section, we will discuss the scope of SOMA (Section 3.1), its underlying foundational commitments (Section 3.2), and how it is organized (Section 3.3).

3.1. Scope

The broad scope of our work is everyday object manipulation tasks in autonomous robot control, and in particular the motion and force characteristics of objects. The research question driving us is whether a single general control program can be written that can generate adequate behavior in many different contexts: for different tasks, objects, and environments. The employment of a general plan thus requires an abstract task and object model, and a mechanism to apply this abstract knowledge in situational context.

A more fine-grained scope is defined through a set of competency questions that are documented in the NEEM-Handbook [4]. Some examples related to the modeling of affordances are “what can an object be used for?” and “what can an object be used with?” (referring to the fact that affordances arise through the meeting of compatible dispositions), as well as “what cannot be used to manifest an affordance?”. Thus, the ontology offers ways to indicate what objects – given semantic knowledge about them – provably can or provably cannot be used for some purpose, with undecided cases being passed on to other mechanisms, e.g. simulation-based testing.

3.2. Foundational Commitments

SOMA is based on the DOLCE+DnS Ultralite (DUL) foundational framework [17]. This decision is greatly motivated by their underlying ontological commitments. Firstly, DUL is not a revisionary model, but seeks to express stands that shape human cognition. It assumes a multiplicative approach. Our work, however, seeks to apply a reductionist approach where possible – rather than capturing, for example, the flexibility of our usage of objects via multiple inheritance in a multiplicative manner, we commit to a reduced *ground* classification and use a *descriptive* approach for handling this flexibility, as provided by the addition of the *Descriptions and Situations* extension of DUL [18]. For this a primary branch of the ontology represents the ground **physical model**, e.g. objects and actions, while a secondary branch represents the **social model**, e.g. roles and tasks. All entities in the social branch are mind-dependent entities, i.e. they constitute social objects that represent concepts about, or descriptions of ground elements.

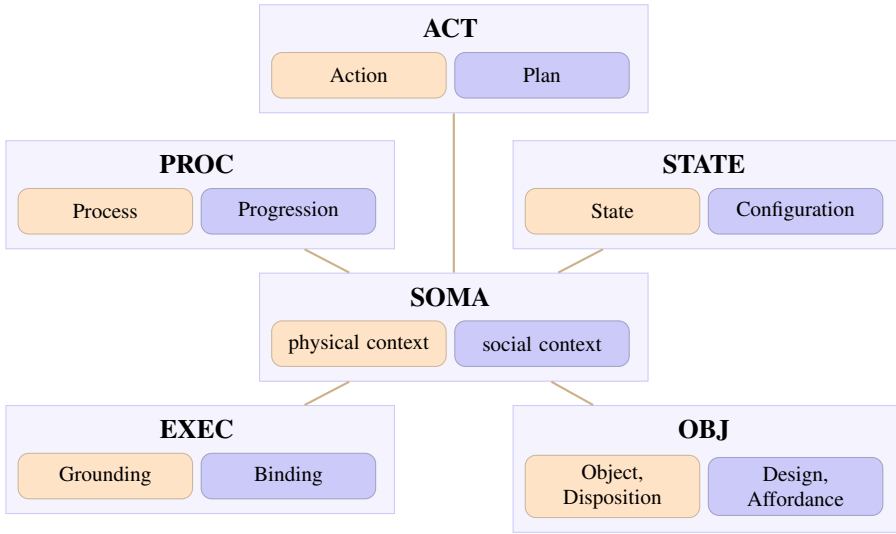


Figure 2. The modular organization of SOMA. Each module defines concepts and relationships used to represent physical (orange) and social (purple) activity context.

Every axiomatization in the physical branch can, therefore, be regarded as expressing some physical context whereas axiomatizations in the descriptive social branch are used to express social contexts. Already some dedicated relations and design patterns are provided that connect both branches⁴. For example, as detailed in Section 4.1, the DUL relation *classifies* connects ground objects, e.g. a hammer, with the roles they can play, while this does not represent classification in the logical sense, we find the distinction between what an object is and what it can be used for very suitable for our approach. Thus, we can state that a hammer can in some context be conceptualized as a murder weapon, a paper weight or a door stopper. Nevertheless, neither its ground ontological classification as a tool will change nor will hammers be subsumed as kinds of door stoppers, paper weights or weapons via multiple inheritance. Following a quick overview of the central modules of SOMA, we will provide detailed examples of where and how our commitments apply in Sections 4 and 5.

3.3. Module Overview

SOMA is organized in several modules that conceptualize different aspects of physical and social activity context (Figure 2). The different modules correspond to different event types (*ACT*, *PROC*, *STATE*), objects that participate in the activity (*OBJ*), and execution context (*EXEC*). The scope of each of these modules is outlined below.

The scope of the *OBJ* module is the representation of physical objects, and their qualities. The module includes two taxonomies used to classify objects: an object taxonomy in the grounded branch, and a role taxonomy in the descriptive branch. It further includes a taxonomy of dispositions to represent the potential of using an object in some way, and a design taxonomy used to categorize objects based on function, structure, and aesthetics. This will be described in more detail in Section 4.

⁴<http://ontologydesignpatterns.org>

The scope of the *ACT*, *PROC* and *STATE* modules is the contextualization of actions, processes and states. Actions are defined as events performed by an agent (physical context), and structured by a plan that is executed by the agent (social context). Plans may further impose constraints on steps in the plan, and objects that may play a role (*EXEC* module of SOMA). An action may cause processes to be started or stopped, and states to be changed. Processes, such as motions, are defined as events considered in their evolution. The difference between a state and a process is that, when considering time slices of the event, for states, these time slices always have the same type as the state (states are homeomeric), but for processes this is not the case. In SOMA, we define a taxonomy of event types in the descriptive branch used to classify actions, and to further decompose them into motion phases, state changes, and physical interactions caused by them. This will be described in more detail in Section 5.

4. Object Representation in SOMA

One of the reasons that everyday activity is a hard problem is the immense amount of variations an unrestricted environment may have, and the resulting potentials of interaction for an agent. Each type of object needs to be handled differently depending on its properties. However, object manipulation tasks are often defined independent of the type of object that is manipulated. It is thus crucial to employ an abstract object model, and a mechanism for applying abstract object knowledge to novel situations.

The main link between objects and actions in SOMA is that objects participate in events in the physical branch, and that the social branch represents the interpretation of their participation. This is elaborated in Section 4.1. Next, in Section 4.2, we will discuss how objects are organized along their design. However, the agent might further need to find suitable candidate objects to perform a task by reasoning about which objects have the potential to be used in a certain way. We employ an object disposition model for that purpose which is discussed in Section 4.3. An example object class is illustrated in Figure 3. In the example, a *Cup* is defined as a type of *DesignedContainer*, and thus inherits qualities through the class membership such as that it has a shape, and that it can be used as a container. However, a cup is specifically designed to contain liquid substances which can be captured by the notion of *FunctionalDesign* in SOMA. The full axiomatization of the concept may contain several similar statements to specify other aspects of object qualities such as that cups afford containment for other object classes too, that they have a specific structural design, etc. Such a definition can be exploited to formulate reasoning queries such as which known objects could be used for a particular purpose, e.g. for storing water, or what the potential uses of an object are. An example of such a reasoning query is provided in Section 6.1.

4.1. Object Types

For the classification of objects we employ the *Role* pattern provided by the foundational layer. Roles are *Concepts* and, as such, reside in the *SocialObject* branch of DUL. For human agents the ascription of roles to entities comes very natural. *He is a student* does not imply an *isa* or *instanceof* relation between some male individual and a student class. It is rather meant that at this point of his life the individual plays the role of a student, which, however, can and will change over time.

<pre> Class: LiquidContainmentDesign SubClassOf: FunctionalDesign isDesignOf only (hasDisposition some (Containment and (affordsTrigger some (classifies only Liquid))) </pre>	<pre> Class: Cup SubClassOf: DesignedContainer isDescribedBy some LiquidContainmentDesign </pre>
-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------

Figure 3. An example of how object classes are represented in SOMA.

This role pattern is of paramount importance, especially in the modeling of affordances discussed in Section 4.3. In the model presented herein, we import the roles that have been established in the field of frame semantics [19]. The selectional restrictions imposed by the *classifies* relation are used in a number of reasoning processes ranging from natural language understanding to tool selection. As certain roles can only classify physical agents or specific types of designed artifacts these axiomatizations provide substantial information about context dependent *meaning* of objects.

4.2. Object Designs

The organization of objects along a taxonomy is difficult as objects can be categorized in many ways. A notion of design is useful to capture object categories corresponding to structural, functional, or aesthetic patterns. Designs are in particular useful to conceptualize *refunctionalized* entities, and to support an agent to *hypothesize* unknown functions served by an entity. For example, a wooden pallet can be reused for the construction of furniture such as sofa, bed etc. The categorization of objects along their design can be employed in order to allow the use of more general plans, where, instead of object types, the plan refers to structure, aesthetics, or function.

Within the scope of SOMA, the *Design* concept belongs to the social branch. SOMA considers *structural*, *functional* and *aesthetic* aspects of design. A design describes classes of objects that host a common design-relevant quality. These qualities are dispositional, geometrical, and aesthetic aspects of the object. This corresponds to our design categorization into functional, structural, and aesthetic design. Each *Design* concept defines restrictions on the corresponding quality type that needs to be fulfilled by any object described by the design. These restrictions can also represent sufficient conditions under which an object is thought to be described by the design which allows the classification of entities given their design pattern can be detected.

In the scope of this work, we only consider functional aspects of objects. These are represented using a model of dispositional qualities which is discussed next.

4.3. Object Dispositions

Objects are important to an agent because they allow it to perform, or prevent it from performing, actions to achieve its goals. The notion of “affordance” was put forth by Gibson as “. . . what it offers the animal, what it provides or furnishes, either for good or ill” [20]. However, though evidently useful as a way to organize actionable knowledge about the world [21], affordances proved very difficult to model ontologically. Several approaches have been proposed, such as regarding affordances as qualities [22] or as

events [23]. Nonetheless, we think these approaches are not satisfactory. Affordances are relational, characterizing a potential interaction of several objects, and therefore should not be treated as either a quality belonging to an object, nor as an event. We do recognize that some qualitative aspects of objects contribute to affordances, however, which is why we constructed our model around the interplay of Turvey’s notion of *disposition* [24] and Gibson’s notion of *affordance*.

In SOMA, the *Disposition* concept is defined as an object quality that allows an object to participate in events that *realize* an affordance. The *Affordance* concept itself, however, is defined as the relational context holding between several objects that play different roles such as being the “bearer”, “trigger”, or “background” of an affordance. Our modeling allows us, via a mixture of DL and other reasoning mechanisms such as simulation, to answer several interesting questions such as what affordances might an object provide in some combination with others, what objects might, or probably would not, be able to provide a given affordance, what combinations of objects would work towards providing an affordance etc. For more details on our disposition and affordance model, we invite the reader to consult our previous work on this topic [25].

5. Event Representation in SOMA

The information gap between an instruction given to an embodied agent and the way it has to move its body to *successfully* execute the instruction is often immense. Consider, for example, a recipe for cooking noodles that contains an instruction to *boil water in a pot*. It is simple to decompose this instruction into several steps with individual sub-goals such as finding pot and tap, placing the pot underneath the tap, and filling the pot with water. However, the more difficult problem is how the agent has to move its body in each step such that the goal is achieved, and unwanted side-effects are avoided. Little variations in motion behavior may have drastic consequences in tasks that require delicate interaction. It is thus essential for agents performing actions in the physical world to reason about *how they should move* to achieve their goals in an appropriate, flexible and robust manner which is an unsolved problem for the general case.

SOMA attempts to support an agent facing this problem by equipping it with knowledge about relationships between abstract descriptions and their realization. The support is twofold. First, the agent may employ more general plans where informational gaps are filled by reasoning over knowledge represented with SOMA. Second, the agent may employ SOMA for understanding and generalizing observations. This means that agents can interact safer in environments with incomplete information, and that they can learn general patterns from specific situations.

An illustrative example of the representation of a pouring plan in OWL Manchester Syntax is provided in Figure 4. The plan is represented as an ABox ontology, i.e. as a collection of facts about the plan: what task it defines, and what steps it describes. Steps are conceptualizations of the events that realize them. They specify the roles objects need to play during the event, and may further specify ordering constraints using Allen’s relations such as that realizations of the step *Pouring0* are *started by* realizations of the *Approaching0* step. Finally, resources may need to be shared among different steps within a plan, for example the source from which is poured (with role *Source0*) is the

<p>Individual: PouringPlan_0 Types: Plan , Description Facts: defines Pouring_0 , hasPhase Approaching_0 , hasPhase Tilting_0</p> <p>Individual: Pouring_0 Types: Task , Concept Facts: usesRole Patient_0 , usesRole Source_0 , usesRole Destination_0 , startedBy Approaching_0</p>	<p>Individual: Approaching_0 Types: Motion Type , Concept Facts: usesRole Destination_1 , overlapsWith Tilting_0</p> <p>Individual: Tilting_0 Types: Motion Type , Concept Facts: usesRole Patient_1</p> <p>Individual: Binding_1 Types: Role Binding , Description Facts: hasBinding Source_0 , hasBinding Patient_1</p>
------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Figure 4. An example of how plans are represented in SOMA.

same entity as the patient during the tilting motion (with role *Patient0*) which is captured through a role binding in the plan definition.

The reason that plans are represented as ABox ontologies in SOMA is that identity constraints cannot be expressed as OWL-DL axioms, i.e. distinct steps of a plan with the same type cannot be defined by different axioms. However, sequencing information can be encoded in the ABox. This has the drawback that an OWL reasoner cannot recognize the plan that was executed by an agent. In general, the machinery necessary to perform, or recognize the execution of a task is outside the scope of OWL-DL. Nonetheless, we have committed to encoding as many constraints on tasks as we can via OWL-DL axioms.

In the following, we will first introduce our hierarchical organization of tasks, processes and states in Section 5.1, and, second, how they are decomposed into phases with explicit goals and individual knowledge pre-conditions in Section 5.2. Finally, we will discuss our modeling of force dynamical characteristics in Section 5.3.

5.1. Event Types

One of the most important demands on a cognitive system is to reason about actions; colloquially speaking, an agent constantly asks itself what to do, and how to do it. This however opens up another question, namely what exactly is the entity that the agent represents – an actual event, or an interpretation of one.

As an example, consider this scenario: a robot moves toward a table carrying a plate. Midway, its gripper releases, dropping the plate, which shatters against the floor. Perhaps the robot had to transport the plate to the table, and it failed to do so; or perhaps it was required to drop the plate as part of some material test, and the table was just there for some other reason. Just by observation of the action, without other interpretive context which includes knowledge of what the robot was told to do, there is no reliable way to tell. The failed transport interpretation does seem more likely a priori, but only because we have more often seen people tell robots to transport plates rather than break them; we still make use of an expected interpretive context.

As a result, we do not define a taxonomy of action events in our ontology, but rather of tasks that are used to conceptualize actions. For example, the *Grasping* concept is defined as task in SOMA, and it is used for the *classification* of events that are interpreted as an intentional grasping activity. This classification pattern between

events and their conceptualization is provided by the foundational layer of SOMA. However, within the foundational layer, this pattern is only instantiated for actions and their conceptualization. In our modeling, motions of an agent and other processes, as well as state events are used to structure an activity. Thus, we also need to represent processes and states in the ground and the descriptive ontology. The same pattern applies: the concepts `Process` and `State` are defined in the ground ontology, and their conceptualization in the descriptive ontology, and a relationship between both branches is established through the aforementioned classification pattern.

5.2. Event Phases

Actions in SOMA are composed of distinct *phases*. Each phase has its individual goal, and requires a different movement strategy to be executed successfully. The phases correspond to different stages of an object manipulation task, usually separated through *contact events*. Flanagan et al. have pointed out the importance of contact events in object manipulation tasks [26]. The authors have shown that contact events cause a distinct pattern in sensory events, and that they can be used as *sensorimotor control points* for aligning and comparing predictions with actual sensory events. Another justification is that humans have shown to direct their gaze to contact points when they perform object manipulation tasks, or when they observe another agent performing a task.

The structure of activities in SOMA is governed by a set of design patterns. At its core, SOMA activity modeling builds on top of the *basic plan* ontology design pattern that represents plans and their execution. The pattern defines that an execution is a situation that *satisfies* the description of the plan. However, the pattern is defined too specific for the scope of this work, as we also want to provide descriptive context for states and processes. Hence, we generalize this pattern such that it can be instantiated for actions, states and processes: An `Action` is described in a `Plan` which is a description having an explicit goal. A plan satisfies situations that include action sequences that match the structure of the plan, such situations are called `Plan Executions`; a `State` is described in a `Configuration` which includes constraints on regions of entities and relationships between them. A configuration satisfies situations in which all constraints of the configuration are satisfied; and a `Process` is described in a `Process Flow` which is a description of the progression of the process. A process flow satisfies situations that include a process that progresses in the described way. Another aspect of activity structure can be captured by SOMA in, what we call, *execution contexts*. These are representations of how different phases of an activity constrain each other depending on conditions encountered in the activity execution. In particular, we define the `Binding` concept as identity constrain representing that a parameter or role grounding is the same in different phases, however potentially being classified differently.

Ordering constraints are expressible in SOMA through a sequence pattern based on Allen's interval calculus [27]. Allen's calculus defines thirteen relations between time intervals including *before*, *after*, *overlaps*, and *meets*. This is useful to, on the one hand, represent precedence of one phase strictly following the other, and, on the other hand, it allows to cope with concurrency in the sequence. We apply this algebra to event types that are defined within the descriptive context of a plan or process flow. However, reasoning about sequences is not well supported in OWL. Instead, we translate interval relations

into a point graph to perform point-based reasoning [28]. Point graphs are directed acyclic graphs where nodes are the endpoints of intervals, and an edge is added for each axiom $a < b$ where a, b are interval endpoints. A non empty path from an endpoint a to another endpoint b further implies that $a < b$ through the transitivity of the relation. Event relations can be inferred through relations between their endpoints. For example, an interval i_1 precedes another interval i_2 iff $e_1 < s_2$ where s_2 is the starting point of i_2 and e_1 is the ending point of i_1 . However, this only covers the *pointisable* subclass of the algebra which means that e.g. disjunction axioms are not expressible.

Knowledge about the structure of activities can be employed by an agent in both directions: for planning an activity, and for interpreting observed events. Planning can be seen as a mapping from the descriptive to the grounded branch of SOMA, while interpretation maps the other way. For embodied agents, planning goes beyond mere decomposition of an activity into steps, the agent may further need to decide what objects it should use, how it should move, with what speed, and how much force it should apply when getting into contact with some object. SOMA may be employed by the agent to find potential sequences of steps and motions to execute a task, to support finding potential objects playing some role during the activity, and to constrain the values of parameters of a task. Interpretation of events, on the other hand, is often possible through detection of contact events, types of motions, and states. These can be used as tokens for an activity parser that uses SOMA as a grammar, this will be described more in Section 6.

Knowledge Pre-Conditions. In order to execute a motion, an agent has to invoke one of its control routines with a set of arguments. Higher-level routines may have a notion of object, but at a lower-level all boils down to numbers such as with what effort the robot moves, how fast, etc. SOMA allows to define constraints for both cases: for the types of objects that can play a role during the action, and for the value of parameters. This is done by using restrictions on what types of objects or regions can be classified by some role or parameter. This information is used to reduce the search space for doing an appropriate object or parameter selection (Section 4.3).

Goals. A goal is a description of a desired situation, and it is achieved only if the situational context, after the execution has been finished, satisfies this description. SOMA is more specific about what it means to execute an action successfully as it decomposes it into processes and states where the goal of the task is that the progression of processes evolves, and that state changes occur as described. Particularly important are the contact states in object manipulation tasks, as they represent control points for the agent when generating or observing behavior.

5.3. Event Force Characteristics

A contact state is an indicator for whether objects are touching each other or not. Patterns of such states are useful for distinguishing between categories of activities. However, different activities may cause the same pattern while their goal is different, or even the opposite of each other. This is, for example, the case for pulling and holding. Both cause the same pattern of an endeffector getting into contact with another object. But the force characteristics are different: the goal of a pulling task is to overcome the inertial force of the object to set it into motion, and the goal of a holding task is to neutralize any external force that would set the object into motion. Another aspect is that an agent performing

such a task needs to decide how much force to apply. In order to make this decision it is valuable to know what the intended force-related consequences are.

SOMA supports the representation of force characteristics using Talmy’s notion of force dynamics [29]. Talmy distinguishes between two entities that participate in force dynamical processes: the *Agonist*, and the *Antagonist*. An agonist is the subject of a force dynamical expression, while the antagonist is the opposing force in the expression. Each expression has an intrinsic force tendency either to set the agonist into motion, or to keep it resting. Whether the tendency can be realized or not depends on which of the two entities is the *stronger* entity.

6. Evaluation of SOMA

SOMA was developed to provide robots with the capability to answer a set of competency questions about everyday activities. Thus, we evaluate SOMA by validating that these questions can be answered (in Section 6.1). Due to space limits, we will only elaborate on selected examples here. The full range of competency questions is documented in the NEEM-Handbook [4]. Furthermore, the relevance of these competency questions can be demonstrated through applications of SOMA and their evaluation. A practical employment of SOMA is demonstrated in the *EASE Robot Household marathon* [30]. Here, we will provide an overview of SOMA applications in prior work to verify its use in the application domain of autonomous robotics (in Section 6.2).

In version 1.1.0, SOMA contains 1330 logical axioms, 416 classes, 203 object properties, and 38 data properties. Its expressivity is SROIQ(D). More metrics are listed on the SOMA webpage ⁵. They are automatically computed when SOMA is deployed through a web service based on OntoMetrics ⁶.

6.1. Reasoning with SOMA

Being written in DL, SOMA can be processed with standard DL reasoners such as HermiT. Because reasoning with the ontology is important during its use, our process of updating the ontology includes a reasoning step as well, also performed with HermiT, to verify that updates do not insert unsatisfiable concepts or empty properties. In more detail, every commit to the SOMA repository triggers subsumption and classification queries, and the discovery of concepts or properties equivalent to `Nothing` triggers a warning. This eases maintenance and scaling up of SOMA while keeping it consistent.

We will next exemplify how the ontology can be reasoned with “at runtime”, during some activity of a robot. Knowledge in SOMA covers, among others, aspects such as dispositions and affordances of objects. A question a robot might have is, what object in its environment could be used for a particular purpose, e.g. to contain some liquid. To this end, the robot will query the ontology by first defining a new “query” concept, formulated in Listing 1 and then ask which of the objects it knows about can be proven to belong to this query concept via a subsumption query – objects that are individuals of subconcepts of the query concept can be used.

⁵<https://ease-crc.github.io/soma/>

⁶<https://ontometrics.informatik.uni-rostock.de>

Sometimes, no known objects might be provably appropriate for a purpose. In such cases, one might try some other methods, such as testing in simulation, but such methods are themselves costly and so a filtering of candidates via reasoning is useful. In this example, the robot might ask, “what cannot be used to contain a liquid”. This is also achieved with the help of the “query” concept illustrated in Listing 1, but in a different manner. For a named concept C present in SOMA, that is a subconcept of `DesignedArtifact`, do a satisfiability query for the intersection of C with the query concept. If this intersection is provably empty, objects that are instances of C need not be tested for the affordance.

Listing 1: A query concept to find objects which can contain liquids

```

Class : WithAffordance_Containment_Liquid
EquivalentTo :
  DesignedArtifact and (hasDisposition some
    (Containment and (affordsTrigger some (classifies only Liquid))))

```

OWL-DL was designed for the representation of encyclopedic knowledge, and has limited scope for domains such as dynamical characteristics. This concerns, for example, reasoning about the temporal ordering of steps that execute a task which we handle through *point-based* reasoning where SOMA is enriched with definitions of temporal relations. Another example is *simulation-based* reasoning for affordance testing [31,32]. Thus, SOMA is used as a common model in a hybrid reasoning framework.

6.2. Applications of SOMA

The applicability of SOMA in the domain of autonomous robotics has been demonstrated in several scientific publications which we will briefly discuss in this section.

Grounding task parameters often requires predictive models which can be trained over instances of successful performance. Such experiential knowledge is in particular useful to learn context-dependent *plan specializations*. That is, how the parameters of the plan can be constrained within the scope of some context to reduce the search space of parameter selection during plan execution. The learning problem is then defined with respect to a contextual pattern, and experiential samples are only considered when their contextualization matches this pattern. We have demonstrated this capability in another work where a robot learns to execute a general fetch and place plan based on experience acquired through the execution of more constrained tasks [33].

Learning mechanisms often require large amounts of training data. One modality for acquisition is observation of other agents. We have developed an activity parser that is used to find possible interpretations for observed patterns of occurrences such as that objects get into contact with each other, or that the state of an object changed. The grammar used by the parser is a library of plans represented using SOMA. In prior work, we have provided more details about how the social context in SOMA can be grounded in data structures of a game engine [34]. The game engine implements an immersive virtual reality environment with photo realistic rendering and state of the art physics engine. Users perform object manipulation tasks while interactions, states, and motions are monitored, and used as tokens by the activity parser.

Our modeling of tasks also helps to disambiguate vague natural language commands a robot may receive. SOMA allows us to model how tasks relate to and depend on one another, and thus define execution contexts containing not just information about a task's parametrization, but also information about what other tasks it should enable. We use such execution contexts to set up simulation scenarios in which to test task executions and thus select among several interpretations of a vague natural language command [31].

7. Conclusion

In this paper, we have proposed SOMA, a novel activity ontology for robotic agents that combines several established ontology design patterns with models of human cognition. The SOMA ontology has been designed to cover a set of competency questions in order to support robot decision making during action execution or observation. This is done by representing physical and social context of an activity, and by establishing relationships between both contextualizations. These representations are used by robotic agents to fill knowledge gaps in general plans applicable to many situations, and to generate context-specific behavior. They are further used for the representation of observed events, and for reasoning about how they are to be interpreted. We believe that such an expressive activity representation is an important vehicle for transforming robots from just performing a task to mastering the corresponding activity.

References

- [1] Lenat DB, Feigenbaum EA. On the thresholds of knowledge. *Artificial Intelligence*. 1991;47(1):185-250. Available from: <http://www.sciencedirect.com/science/article/pii/0004370291900550>.
- [2] Anderson JE. *Constraint-Directed Improvisation for Everyday Activities*. The University of Manitoba (Canada); 1995. AAINN99082.
- [3] Beetz M, Beßler D, Haidu A, Pomarlan M, Bozcuoglu AK, Bartels G. KnowRob 2.0 – A 2nd Generation Knowledge Processing Framework for Cognition-enabled Robotic Agents. In: *International Conference on Robotics and Automation (ICRA)*. Brisbane, Australia; 2018. p. 512-9.
- [4] Beetz M, Beßler D, Koralewski S, Mihai P, Vyas A, Hawkin A, et al. *NEEM Handbook*; 2020. Accessed: 2021-04-28. <https://ease-crc.github.io/soma/owl/1.1.0/NEEM-Handbook.pdf>. Available from: <https://ease-crc.github.io/soma/owl/current/NEEM-Handbook.pdf>.
- [5] Olivares-Alarcos A, Beßler D, Khamis A, Goncalves P, Habib M, Bermejo J, et al. A Review and Comparison of Ontology-based Approaches to Robot Autonomy. *The Knowledge Engineering Review*. 2019;34:e29.
- [6] Patel R, Hedelind M, Lozan-Villegas P. Enabling robots in small-part assembly lines: The "ROSETTA approach" - an industrial perspective. In: *ROBOTIK*. VDE-Verlag; 2012. .
- [7] Stenmark M, Malec J, Nilsson K, Robertsson A. On Distributed Knowledge Bases for Robotized Small-Batch Assembly. *IEEE Transactions on Automation Science and Engineering*. 2015;12(2):519-28.
- [8] Balakirsky S. Ontology based action planning and verification for agile manufacturing. *Robotics and Computer-Integrated Manufacturing*. 2015;33(Supplement C):21-28. Special Issue on Knowledge Driven Robotics and Manufacturing.
- [9] Polydoros AS, Großmann B, Rovida F, Nalpantidis L, Krüger V. Accurate and Versatile Automation of Industrial Kitting Operations with SkiROS. In: *Towards Autonomous Robotic Systems - 17th Annual Conference (TAROS)*; 2016. p. 255-68.
- [10] Kootbally Z, Schlenoff C, Lawler C, Kramer T, Gupta SK. Towards Robust Assembly with Knowledge Representation for the Planning Domain Definition Language (PDDL). *Robot Comput-Integr Manuf*. 2015 Jun;33(C):42-55.

- [11] Perzylo A, Somani N, Profanter S, Kessler I, Rickert M, Knoll A. Intuitive instruction of industrial robots: Semantic process descriptions for small lot production. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS); 2016. p. 2293-300.
- [12] Tenorth M, Beetz M. Representations for robot knowledge in the KnowRob framework. Artificial Intelligence. 2015.
- [13] Lenat D. CYC: A Large-Scale Investment in Knowledge Infrastructure. Communications of the ACM. 1995 November;38(11):33-8.
- [14] Schlenoff C, Prestes E, Madhavan R, Goncalves P, Li H, Balakirsky S, et al. An IEEE standard ontology for robotics and automation. In: IEEE Int. Conf. on Intelligent Robots and Systems (IROS); 2012. p. 1337-42.
- [15] Prestes E, Carbonera JL, Fiorini SR, Jorge VAM, Abel M, Madhavan R, et al. Towards a core ontology for robotics and automation. Robotics and Autonomous Systems. 2013;61(11):1193-1204. Ubiquitous Robotics.
- [16] Fiorini SR, Carbonera JL, Gonçalves P, Jorge VAM, Rey VF, Haidegger T, et al. Extensions to the Core Ontology for Robotics and Automation. Robot Comput-Integr Manuf. 2015 Jun;33(C):3-11.
- [17] Masolo C, Borgo S, Gangemi A, Guarino N, Oltramari A. WonderWeb Deliverable D18 Ontology Library (final). IST Project 2001-33052 WonderWeb: Ontology Infrastructure for the Semantic Web; 2003.
- [18] Gangemi A, Mika P. Understanding the Semantic Web through Descriptions and Situations. In: On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE - OTM Confederated International Conferences, CoopIS, DOA, and ODBASE 2003, Catania, Sicily, Italy, November 3-7, 2003; 2003. p. 689-706. Available from: https://doi.org/10.1007/978-3-540-39964-3_44.
- [19] Baker CF, Fillmore CJ, Lowe JB. The Berkeley FrameNet Project. In: Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics - Volume 1. ACL '98/COLING '98. USA: Association for Computational Linguistics; 1998. p. 86-90. Available from: <https://doi.org/10.3115/980845.980860>.
- [20] Gibson JJ. The Ecological Approach to Visual Perception. Psychology Press Classic Editions; 1979.
- [21] Yamanobe N, Wan W, Ramirez-Alpizar IG, Petit D, Tsuji T, Akizuki S, et al. A Brief Review of Affordance in Robotic Manipulation Research. Journal of the Robotics Society of Japan. 2018 01;36:327-37.
- [22] Ortmann J, Kuhn W. Affordances As Qualities. In: Proceedings of the 2010 Conference on Formal Ontology in Information Systems: Proceedings of the Sixth International Conference (FOIS 2010). Amsterdam, The Netherlands, The Netherlands: IOS Press; 2010. p. 117-30.
- [23] Moralez LA. Affordance ontology: towards a unified description of affordances as events. Res Cogitans. 2016;7(1):35-45.
- [24] Turvey MT. Ecological foundations of cognition: Invariants of perception and action. American Psychological Association. 1992.
- [25] Beßler D, Porzel R, Pomarlan M, Beetz M, Malaka R, Bateman JA. A Formal Model of Affordances for Flexible Robotic Task Execution. In: ECAI. vol. 325 of Frontiers in Artificial Intelligence and Applications. IOS Press; 2020. p. 2425-32.
- [26] Flanagan JR, Bowman MC, Johansson RS. Control strategies in object manipulation tasks. Current Opinion in Neurobiology. 2006.
- [27] Allen JF. Maintaining knowledge about temporal intervals. In: In Communications of the ACM, Volume 26 Issue 11; 1983. .
- [28] Zaidi AK, Wagenhals LW. Planning temporal events using point-interval logic. Mathematical and Computer Modelling. 2006;43(9):1229-53. Optimization and Control for Military Applications.
- [29] Talmy L. Toward a Cognitive Semantics. Volume 2: Typology and Process in Concept Structuring. Language, Speech, and Communication. Cambridge, MA: MIT Press; 2000.
- [30] Kazhoyan G, Stelter S, Kenfack FK, Koralewski S, Beetz M. The Robot Household Marathon Experiment. In: IEEE International Conference on Robotics and Automation (ICRA). Xi'an, China; 2021. Available from: <https://arxiv.org/abs/2011.09792>.
- [31] Bateman J, Pomarlan M, Kazhoyan G. Embodied Contextualization: Towards a Multistratal Ontological Treatment. Applied Ontology. 2019;14(4):379-413.
- [32] Pomarlan M, Bateman J. Embodied Functional Relations: a Formal Account Combining Abstract Logical Theory with Grounding in Simulation. In: Proceedings of the 11th International Conference on

Formal Ontology in Information Systems. FOIS; 2020. .

- [33] Koralewski S, Kazhoyan G, Beetz M. Self-specialization of General Robot Plans Based on Experience. *Robotics and Automation Letters*. 2019.
- [34] Haidu A, Beßler D, Bozcuoglu AK, Beetz M. KNOWROB-SIM — Game Engine-enabled Knowledge Processing for Cognition-enabled Robot Control. In: *International Conference on Intelligent Robots and Systems (IROS)*. Madrid, Spain: IEEE; 2018. p. 4491-8.